**GitHub Runners**

**Introduction**

To execute a GitHub Actions workflow, a system (virtual or physical) is required to run the jobs. There are two main types of runners:

* GitHub-Hosted Runners
* Self-Hosted Runners

**GitHub-Hosted Runners**

GitHub-hosted runners are virtual machines provided by GitHub. Each workflow runs on a fresh virtual machine instance.

**Supported Operating Systems**

* Ubuntu Linux
* Windows Server
* macOS

**Usage Example**

To specify the runner in a workflow:

runs-on: ubuntu-latest

**Available Images**

* GitHub provides SBOMs (Software Bill of Materials) for each image.
* Example: ubuntu-24.04 includes many pre-installed tools.

**Installing Additional Software**

Even though many tools are pre-installed, you can install custom packages as needed:

jobs:

update-env:

runs-on: ubuntu-24.04

steps:

- name: Install packages

run: |

sudo apt update

sudo apt install <package-name> -y

**Self-Hosted Runners**

Self-hosted runners are infrastructure managed by the user or organization.

**Scoping Options**

* Repository Level
* Organization Level
* Enterprise Level

**Documentation**

Refer to the official GitHub documentation on self-hosted runners.

**Autoscaling Self-Hosted Runners**

Organizations using Kubernetes can automate runner scaling using the Actions Runner Controller (ARC).

**Key Concepts**

* ARC is a Kubernetes operator that manages the lifecycle of self-hosted runners.
* **Scale Sets**: Groups of runners managed by ARC that automatically scale based on job demand.
* ARC can integrate with various cloud environments, including AWS.

**Resources**

* Setup guide: [Actions Runner Controller GitHub](https://github.com/actions/actions-runner-controller)
* AWS-specific setup: [AWS ARC Guide](https://github.com/actions/actions-runner-controller/blob/master/docs/aws.md)

**GitHub Actions Marketplace**

Browse and use reusable actions directly from the GitHub Marketplace.

**Popular Core Actions**

* actions/checkout
* actions/setup-node
* actions/setup-python
* actions/setup-java
* actions/setup-go
* actions/cache
* actions/upload-artifact
* actions/download-artifact
* actions/github-script

**Java Application Workflow Example**

You can define a GitHub Actions workflow to build and test a Java application (e.g., Spring PetClinic).

**Example Workflow Syntax**

name: Java CI

on: [push, pull\_request]

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout source code

uses: actions/checkout@v3

- name: Set up Java 17

uses: actions/setup-java@v3

with:

java-version: '17'

distribution: 'temurin'

- name: Build using Maven

run: mvn clean install

- name: Publish Test Results

if: always()

run: mvn surefire-report:report